



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

*Am*

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|-----------------|-------------|----------------------|---------------------|------------------|
| 09/838,552      | 04/18/2001  | Boris A. Babaian     | 020181004900        | 8760             |

20350 7590 06/03/2005

TOWNSEND AND TOWNSEND AND CREW, LLP  
TWO EMBARCADERO CENTER  
EIGHTH FLOOR  
SAN FRANCISCO, CA 94111-3834

EXAMINER

STEELMAN, MARY J

ART UNIT PAPER NUMBER

2191

DATE MAILED: 06/03/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

# Office Action Summary

|                  |                |  |
|------------------|----------------|--|
| Application No.  | Applicant(s)   |  |
| 09/838,552       | BABAIAN ET AL. |  |
| Examiner         | Art Unit       |  |
| Mary J. Steelman | 2191           |  |

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

## Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

## Status

- 1)  Responsive to communication(s) filed on 02 December 2004.
- 2a)  This action is FINAL.                      2b)  This action is non-final.
- 3)  Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

## Disposition of Claims

- 4)  Claim(s) 35-56 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5)  Claim(s) \_\_\_\_\_ is/are allowed.
- 6)  Claim(s) 35-56 is/are rejected.
- 7)  Claim(s) \_\_\_\_\_ is/are objected to.
- 8)  Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

## Application Papers

- 9)  The specification is objected to by the Examiner.
- 10)  The drawing(s) filed on 18 April 2001 is/are: a)  accepted or b)  objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11)  The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

## Priority under 35 U.S.C. § 119

- 12)  Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a)  All    b)  Some \*    c)  None of:
1.  Certified copies of the priority documents have been received.
2.  Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
3.  Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

## Attachment(s)

- 1)  Notice of References Cited (PTO-892)
- 2)  Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3)  Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)  
    Paper No(s)/Mail Date \_\_\_\_\_.
- 4)  Interview Summary (PTO-413)  
    Paper No(s)/Mail Date \_\_\_\_\_.
- 5)  Notice of Informal Patent Application (PTO-152)
- 6)  Other: Accepted copy of FIGs 1, 4, 5.

RD

### DETAILED ACTION

1. This Office Action is in response to Amendments and Remarks received 21 December 2004. Per Applicant's request, claims 1-34 are canceled. New claims 35-56 have been added. The Specification and Abstract have been amended. New Replacement drawings for FIGs. 1, 4, & 5 have been received. A substitute oath has been received. It is noted that a Request to Correct Inventorship petition and a signed oath was received 2 December 2004.

#### *Drawings*

2. Replacement Sheets Drawings have been received for FIGURE 1, FIGURE 4, and FIGURE 5. In view of the amended drawings, the prior objections to Figures 1 & 5 are hereby withdrawn. In view of the amendment to the Specification, the prior objection to FIGURE 2 is hereby withdrawn.

3. Figure 3 is objected to as failing to comply with 37 CFR 1.84(p)(5) because they do not include the following reference character(s) mentioned in the description:

FIGURE 3, see Specification page 11, line 3, 'very long instruction word format 300'. 300 is not in the drawing.

Corrected drawing sheets are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The

Art Unit: 2191

replacement sheet(s) should be labeled "Replacement Sheet" in the page header (as per 37 CFR 1.84(c)) so as not to obstruct any portion of the drawing figures. If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

### *Specification*

4. In view of the amendments to the Specification and the Abstract, the prior objections are hereby withdrawn.

### *Claim Objections*

5. In view of the cancellation of claims 1-34, all prior claims objections are hereby withdrawn.

6. Claim 49, line 11 recites, "...corresponding executed optimized binary translated code...", should be --...corresponding to executed optimized binary translated code...-- Add 'to' after 'corresponding.'

Claim 56, line 2 recites "...a set of tag values that...", should be --a set of tag values that...-- Add a space between 'values' and 'that.'

Appropriate correction is required.

***Claim Rejections - 35 USC § 112***

7. In view of the cancellation of claims 26 and 28, the prior 35 USC 112 2<sup>nd</sup> paragraph rejections are hereby withdrawn.

***Claim Rejections - 35 USC § 102***

8. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371(c) of this title before the invention thereof by the applicant for patent.

The changes made to 35 U.S.C. 102(e) by the American Inventors Protection Act of 1999 (AIPA) and the Intellectual Property and High Technology Technical Amendments Act of 2002 do not apply when the reference is a U.S. patent resulting directly or indirectly from an international application filed before November 29, 2000. Therefore, the prior art date of the reference is determined under 35 U.S.C. 102(e) prior to the amendment by the AIPA (pre-AIPA 35 U.S.C. 102(e)).

9. Claims 35-56 are rejected under 35 U.S.C. 102(e) as being anticipated by US Patent 5,832,205 to Kelly et al.

Per claim 35.

A binary translation system, comprising:

Art Unit: 2191

Kelly: Col. 10, line 59 – col. 11, line 6, describes a hardware processing portion (system) that ‘morphs code’ (binary translation).

-a non-optimizing foreign code execution module configured to maintain dedicated foreign state for each foreign binary operation executed allowing for the exceptions arisen to be handled precisely;

Kelly: Col. 11, lines 4-6, explains that exceptions and errors are handled to maintain correct state. Col. 12, lines 3-58, disclose a technique to store ‘uncommitted’ updates, and rollback to a ‘dedicated foreign state’ upon detection of an exception.

-an optimizing binary translator configured to translate foreign binary operations into optimized sequences of host operations in such a way as to improve the speed of execution of the sequences;

Kelly, col. 14, line 54 discloses ‘optimizing’ translated code. Also see col. 15, lines 21-30 which discloses optimizing to accelerate processing.

-a host CPU configured to execute the host operations;

Kelly: See FIG. 2, col. 14, lines 3-18 which shows a ‘morph host hardware’ running translated instructions (configured to execute the host operations).

-...to generate...for optimized sequences of host operations,

Art Unit: 2191

Kelly: Col. 14, lines 50-60, discloses that translator optimizes the operations (optimizes sequences of host operations), reorders and schedules.

-...describes operations required to calculate a corresponding foreign state for the appointed points;

Kelly: col. 14, lines 52-53, "...converts those target instructions to the primitive host instructions capable of execution (describes operations required to calculate corresponding state)..."

-... track an executable path of optimized sequences of host operations and record host operation addresses at appointed points of the host operation sequences being executed;

Kelly: Kelly disclosed at col. 12, lines 12-14, an emulator that tracks the target state at all times.

Col. 15, lines 6-11, describe tracking an executable path. Col. 17, lines 27-36 disclose that a successful execution of a code segments results in memory stores, held in a store buffer, being written to memory(committing executed host operations by recording host values at memory addresses.)

-a recovery mechanism configured...to continue foreign codes execution based on information in the documentation in case of the exception arisen during the execution of the corresponding optimized host codes.

Kelly: Col. 17, lines 47-58 discloses a recovery mechanism used if an exception is encountered.

It is inherent that 'documentation' is used to allow the code to continue, as a recovery method

Art Unit: 2191

provides for dumping gated store buffer values and reloading correct values from an 'official register.'

Kelly failed to explicitly disclose 'a documentation tracker configured to track an executable path...' and "a documentation configured to generate / describe operations required..."

However, Kelly does disclose code morphing software that handles exceptions and errors (col. 11, lines 3-7), by replacing working state with correct target state as necessary. It is inherent that some type of 'documentation' is kept by the executing system to track successful / unsuccessful execution of code segments, used to evaluate and correct binary translations.

Per claim 36.

-the sequence of host operations consumes input values from the dedicated foreign state and puts output results into the said foreign state.

Kelly: As an example, col. 13, lines 51-57, the state saving and restoring mechanisms causes the target (host) state to be restore to its most recent official version (consume input values from the dedicated foreign state, restore last known correct values). As an example, col. 16, lines 56-63, disclose a commit operation (puts output results into foreign state) whereby instructions executed without generating an exception cause working register values to be transferred to the official target registers (put output to foreign state).

Per claim 37.



Art Unit: 2191

-the dedicated foreign state is not changed in case of exception arisen during execution of the host operations sequence.

Kelly: Col. 17, lines 47-49 discloses that upon detection of an exception, the calculated values in the gated store buffer are dumped (not used to change foreign state).

Per claim 38.

-a foreign exception handler is executed in response to the exception occurring during execution of the host operations sequence, the foreign exception handler configured to take input values from the unchanged dedicated foreign state.

Kelly: Col. 17, lines 47-54 discloses that foreign exception handler causes values retained in the official registers (foreign registers) to be placed back into the working registers (of host) upon detection of exception during execution of host operations sequence.

Per claim 39.

-the optimizing translator is configured to combine all host operations sequences corresponding to a number of foreign binary operations into one executable unit.

Kelly: Col. 14, lines 19-38 disclose that a VLIW (host operation) may combine multiple foreign (target) operations. Also, see col. 16, lines 28-31, which discloses a plurality of primitive operations may be included in a translated (host) instruction.

Per claim 40.

Art Unit: 2191

-the optimizing translator removes intermediate outputs from the dedicated foreign state for each sequence of host operations and saves information about the intermediate outputs in the documentation.

Kelly: Col. 13, lines 9-11, discloses that intermediate outputs are saved in a 'gated store buffer' documentation.

Per claim 41.

-the documentation describes a virtual foreign state between each sequence of host operations.

Kelly: Col. 16, lines 34-63 discloses how the morph host program (host) executes instructions and retains a 'virtual foreign state' until it is determined that correct execution has occurred (between each sequence of host operations).

Per claim 42.

-the optimizing translator preserves values from each virtual state in temporary registers without re-using them until next virtual foreign state is passed.

Kelly: Col. 16, lines 47-49, "...working registers hold values (preserves values) determined by the primitive operations of the translated instructions (virtual foreign state) executed to that point." Col. 16, lines 59-63 explain that the values are held in the working registers (without re-using) until it is determined that they are correct (until next virtual foreign state is passed) and committed.

Per claim 43.

Art Unit: 2191

-every memory write operation changes unconditionally the dedicated foreign state and is considered as a possible recovery point in case of exception.

Kelly: Col. 20, lines 57-61, "Without a means to distinguish memory from memory mapped I/O, it is necessary to treat all memory with the conservative assumptions..." Kelly disclosed issues with translating memory access operations, and suggested an improved technique using the A/N bit. These instructions comprise a 'set of target instructions', or a Recovery Point. Col. 21, lines 49-53, "...uses the A/N bit to determine whether a failure of speculation has occurred as to whether an access into memory or a memory-mapped I/O device...", col. 22, lines 3-17, "...a set of instructions may first be translated as though it were to affect memory...(translate, detect errors, commit and repeat)."

Per claim 44.

-the recovery point corresponds to one of the virtual foreign states as described in the documentation.

Kelly: Col. 16, lines 56-63 discloses that working register values (virtual foreign states) are transferred 'at the end of the set of instructions' (at a recovery point). Values are committed at the end of the set of instructions / at a recovery point / a commit point if execution does not produce an exception.

Per claim 45.

-the documentation is a host to foreign registers computation rules table or host binary code used to recompute the foreign state.

Art Unit: 2191

Kelly: Col. 16, lines 45-50 discloses that 'official target registers' hold values to recomputed foreign state in the case of encountering an exception. Host binary code, (executing morph host program) that encounters an exception includes 'documentation' to replace working state (col. 11, lines 3-7).

Per claim 46.

-the host CPU is configured to execute host operations and produces exceptions as if it were a foreign CPU.

Kelly: Col. 10, lines 62, "emulating software portion" – host CPU produces exceptions as if it were a foreign CPU. It emulates the foreign CPU.

Per claim 47.

-documentation tracker is a special host operation, a recovery point register and an apparatus for saving instruction address in the recovery point register in response to executing the recovery point operation.

Regarding a 'documentation tracker', see rejection of claim 35 above. Kelly disclosed, col. 16, lines 56-63, whereby 'working register values' are transferred to the official target registers (recovery point register) when 'documentation' determines that host operations have been correctly executed. The instruction address in the recovery point register is saved by a 'gated store buffer' (col. 17, lines 1-6) and are written to memory / committed (at a saved instruction address) (col. 16, lines 19-20) when it is determined that a correct execution has occurred.

Art Unit: 2191

Per claim 48.

-the recover mechanism is a host program configured to recompute a correct dedicated foreign state according to the documentation in the case of an exception generated.

Kelly: Col. 12, lines 59-60 disclose that updates (correct dedicated foreign state) occur on target instruction boundaries. Col. 13, lines 1-4, uncommitted memory stores may be dumped (a recovery mechanism) when an exception is detected, whereby correct dedicated foreign state may be recomputed (col. 13, lines 51-57).

Per claim 49.

A method of recomputing a dedicated foreign state in a binary translation system from documentation generated by an optimizing translator in a case of an exception arising during execution of optimized binary translated code translated from a foreign code, the method comprising:

Kelly: Col. 1, line 11, "...methods..." Col. 13, lines 51-57 discloses recomputing a dedicated foreign state from 'documentation' (state saving and restoring mechanisms) generated by an optimizing translator.

-designating a set of recovery points in the optimized binary translated code during optimized translation of the foreign code, wherein each recovery point represents a foreign state;

Kelly: Col. 12, lines 59-60 discloses designating a set of recovery points (updates chosen to occur on integral target instruction boundaries) Official memory represents foreign state and

Art Unit: 2191

memory stores and working register state are committed to foreign state upon successful execution of optimized binary translated code.

-generating a set of documentations during the optimized translation of the foreign code...

Kelly: Col. 13, lines 32-36 discloses 'code morphing software' (documentation) provides for saving state during translation of foreign code (Col. 2, lines 42-54- optimizing translation of foreign code).

-wherein each documentation in the set of documentations corresponds to a recovery point in the optimized binary translated code and describes operations required to calculate a corresponding foreign state for the recovery point;

Kelly: Col. 11, lines 25-67 describes 'code morphing software' (produces 'documentation' / logic related to recovery points and translated code). Col. 12, lines 59-60 disclose that integral target instruction boundaries are likely recovery points, where values are committed prior to executing a subsequent code segment.

-using one of the documentations in the set of documentations corresponding [to] executed optimized binary translated code when an exception arises during its execution to recover a foreign state corresponding to a recovery point for the exception.

Kelly: Col. 12, lines 45-48 discloses a 'rollback' operation whereby recovery allows 'official target (foreign) registers to transfer contents back to working registers (recover foreign state) at prior update / committed location (recovery point) (col. 12, lines 54-58).

Art Unit: 2191

Kelly failed to explicitly disclose 'documentation.' However, Kelly does disclose code morphing software that handles exceptions and errors (col. 11, lines 3-7), by replacing working state with correct target state as necessary. It is inherent that some type of 'documentation' is kept by the executing system to track successful / unsuccessful execution of code segments, used to evaluate and correct binary translations.

Per claim 50.

-the documentation describes a virtual foreign state and is discovered by using a recovery point register.

Kelly: Virtual foreign state is held in the 'working registers' (col. 12, lines 37-58). Col. 12, lines 37-40 explain that registers are maintained on the host (with virtual states, values) and on the target program (foreign code) (to hold committed / recovery point information) (a recovery point register).

Per claim 51.

-the optimizing translator sets recovery point operations in the host code in such a way that there is no unconditional changes in the dedicated foreign state between adjacent recovery points.

Per claim 52.

-detecting an exception in the optimized binary translated code;

Art Unit: 2191

Kelly: As an example, discloses detecting exception at col. 11, lines 2, "...when an exception or error occurs..."

-reading the recovery point register, finding the appropriate documentation in the set of documentations corresponding to contents of the recovery point register;

Kelly disclosed 'appropriate documentation' corresponding to contents of the recovery point register in that the software allows for (col. 12, lines 45-47) a rollback to quickly transfer the content of all official target registers back to their working register equivalents. Translation code / documentation code is present to find the corresponding contents of the recovery point register.

-recomputing a foreign state according to the documentation;

As an example, Kelly disclosed recomputing a foreign state at col. 13, lines 5-9, whereby target (foreign) state is retranslated.

-switching execution of the foreign code to non-optimizing execution;

Kelly disclosed at Col. 13, lines 5-9 foreign code retranslated one at a time and executed in serial sequence (non-optimizing execution)...

-executing each foreign code operation in sequence;

Kelly disclosed at Col. 13, lines 5-9 foreign code retranslated one at a time and executed in serial sequence (in sequence)...



Art Unit: 2191

-updating a dedicated foreign state at the end of each foreign code operation;

Kelly disclosed "updating dedicated foreign state" at col. 12, lines 59-60, to occur on integral target instruction boundaries (at the end of each foreign code operation).

-detecting an exception during the course of non-optimizing execution;

Kelly disclosed at col. 13, lines 1-67 executing non-optimized code and potential exceptions that may be encountered.

-transferring contents of the dedicated foreign state to the foreign exception handler.

As an example, Kelly disclosed (col. 18, lines 47-49) that a separate target memory (foreign memory) may hold the original memory state which is then utilized to replace overwritten memory if a roll back occurs (a foreign exception handler operation).

Per claim 53.

-a recovery point register; and means for saving a host instruction address into the recovery point register.

Kelly disclosed address generation / saving instructions at col. 27, lines 53-59. The recovery point register / official target (foreign) registers (col. 16, lines 56-63) can map a host instruction address.

Per claim 54.

-the recovery register is a dedicated host register.

Art Unit: 2191

Kelly disclosed at col. 18, lines 35-40 that dedicated host register may record data sufficient to recover state of the target computer...in the case of an exception necessitating a rollback.

Per claim 55.

-the means for saving a host instruction address into the recovery point register comprises a dedicated host microprocessor instruction.

Kelly disclosed instruction generation / saving at col. 27, lines 53-59. The morph host method provides for (col. 12, lines 38-48) a set of host or working registers and a set of target registers to hold the official state. The target registers are connected to their working register equivalents through a dedicated interface (instructions exist that map host to foreign registers) that allows an operation called commit to quickly transfer the contents of all working registers to official target registers (recovery point register) and allow an operation called rollback to quickly transfer the content of all official target registers back to their working register equivalents (using values from recovery point register)....

Per claim 56.

-the host instruction is stored in the host microprocessor memory together with a set of tag values that are associated with the host microprocessor instruction,

As an example, Kelly disclosed (col. 18, lines 49-52) a memory rollback that logs each store and the memory data replaced as execution occurs which can be reversed if a roll back is required.

Art Unit: 2191

-wherein the means for saving a host instruction address into the recovery point register comprises a specific tag value associated with the host microprocessor instruction being executed.

Kelly disclosed (col. 27, lines 53-59) generating / saving addresses. It would be inherent that if the embodiment, as noted above, involving a log of each store and memory data replaced, would include some mapping technique, such as a tag value to associate instructions and addresses.

### *Response to Arguments*

10. Applicant's arguments with respect to claims 1-34 have been considered but are moot in view of the new grounds of rejection. Applicant has canceled all previous claims. New claims 35-56 are examined above.

### *Conclusion*

11. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO MONTHS** of the mailing date of this final action and the advisory action is not mailed until after the end of the **THREE-MONTH** shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event,

Art Unit: 2191

however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.


12. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mary Steelman, whose telephone number is (571) 272-3704. The examiner can normally be reached Monday through Thursday, from 7:00 AM to 5:30 PM. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached at (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Mary Steelman



05/26/2005

*Chameli C. Das*  
**OHAMELI C. DAS**  
**PRIMARY EXAMINER**

*5/31/05*



REPLACEMENT

Accepted by Fomin 5.25.05 118

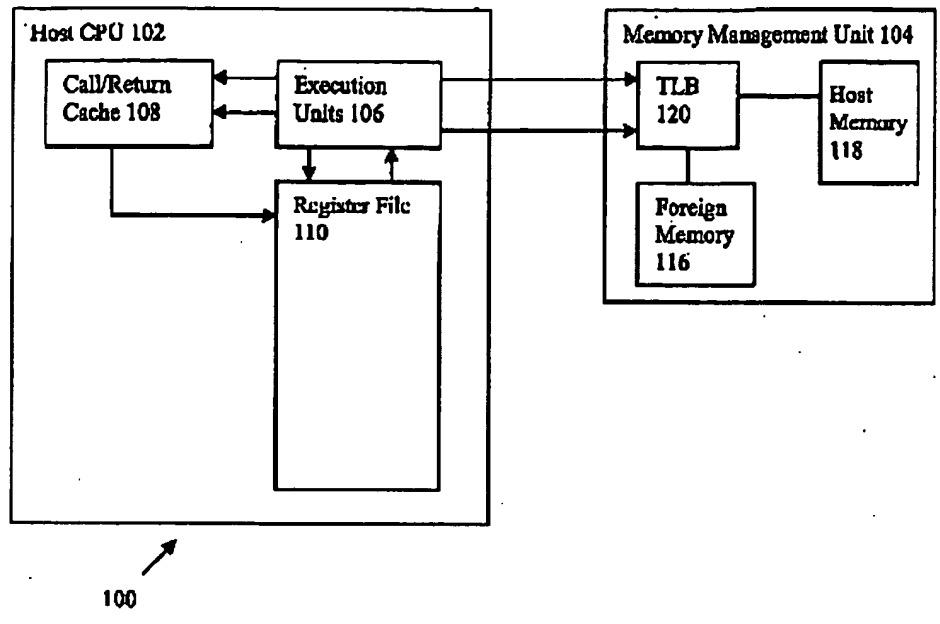


FIGURE 1

Accepted by Examiner 5-25-05 MS



REPLACEMENT

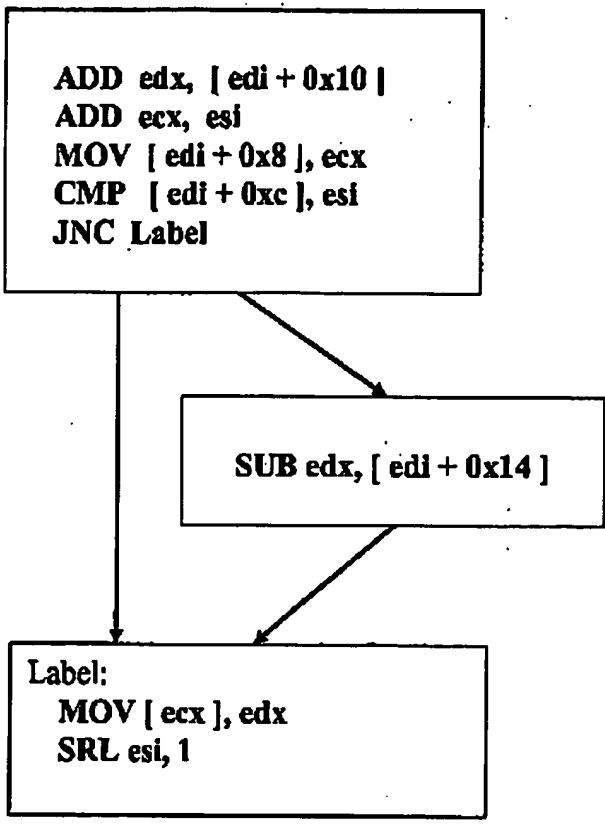


FIGURE 4

REPLACEMENT



Accepted by Examin 5-25-05 MJ

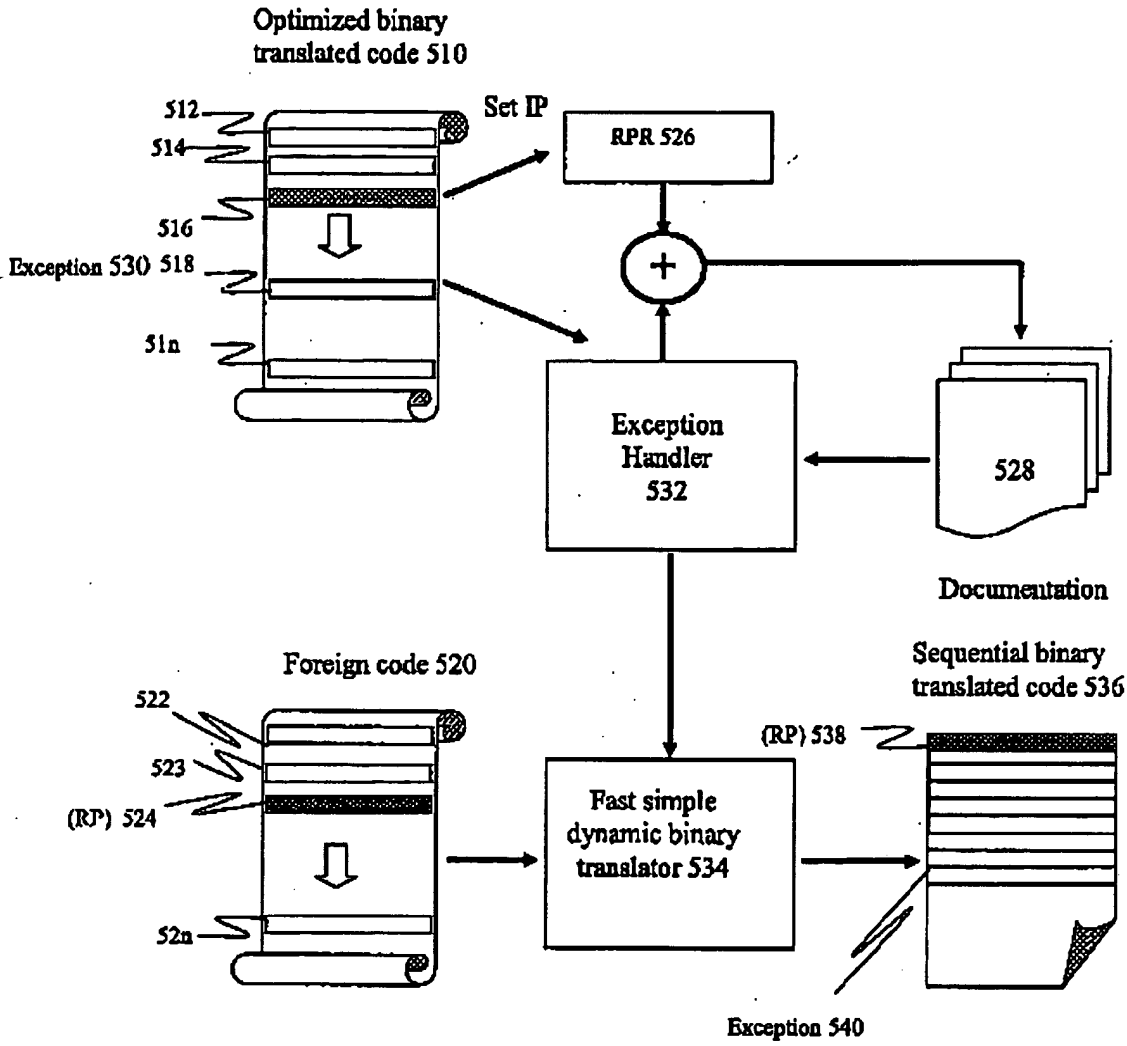


FIGURE 5

2005/12/02  
PAGES 1-8 OF 31  
2005/12/02 10:00 AM